***IN THE CLAIMS*:**

Please **cancel claim 23** without prejudice or disclaimer; **amend claims 1, 3, 6-10, 13, 16, 17, 19, 20, and 22** as indicated below; and **add claims 24-27** as follows:

1. (*Currently amended*) A method of determining a value for a function, comprising:

establishing an n-dimensional lattice having a plurality of lattice points, the function having values at the lattice points, wherein n is a positive integer greater than or equal to **[[**two**]]** 2;

recording values for a subset of the lattice points, the lattice points of the subset being known value lattice points; and

establishing a value for a given lattice point by determining the values of only m of (n+1) known value lattice points defining an n-simplex touching or enclosing the given lattice point, wherein m is a positive integer equal to the number of n-simplexes of non-zero volume whose vertices consist of the given lattice point and n of the (n+ 1) known value lattice points, and by returning a weighted average of said m of the known value lattice points.

2. (*Previously presented*) The method as claimed in claim **1**, wherein n=3 and the n-simplex comprises a tetrahedron.

3. (*Currently amended*) The method as claimed in claim **2**, wherein:

a weighted average of all four known value lattice point values is used if the given lattice point is enclosed by the tetrahedron but is not touched by a face of the tetrahedron,

a weighted average of three of the four known value lattice point values is used if the given lattice point is on a face of the tetrahedron bounded by the three of the four known value lattice points but is not touched by an edge of the tetrahedron,

a weighted average of two of the four known value lattice point values is used if the given lattice point is on an edge of the tetrahedron bounded by the two of the four known value lattice points but is not at a vertex of the tetrahedron, and

~~wherein~~ a value of one of the known value lattice points is used if the given lattice point is also the known value lattice point.

4. (*Cancelled*)

5. (*Cancelled*)

6. (*Currently amended*) The method as claimed in claim **3**, wherein if the given lattice point is enclosed by the tetrahedron but is not touched by a face of the tetrahedron, and the tetrahedron has vertices of known value lattice points with positions A, B, C, D and values a, b, c, d at the respective vertices, and wherein the given lattice point has position P and wherein the volume between four positions is

expressed as Vol (position 1 position 2 position 3 position 4), the value p returned is given by:

p=(Vol(ABCP).d+Vol(ABDP).c+Vol(ACDP).b+Vol(BCDP).a)/Vol(ABCD).

7. (*Currently amended*) The method as claimed in claim **3**, wherein if the given lattice point is on a face of the tetrahedron bounded by the three of the four known value lattice points but is not touched by an edge of the tetrahedron, the three of the four known value lattice points being A, B and C with values a, b and c respectively, the value p returned is given by:

p=((Area(BCP).a)+(Area(ACP).b)+(Area(ABP).c)/Area(ABC).

8. (*Currently amended*) The method as claimed in claim **3**, wherein if the given lattice point is on an edge of the tetrahedron bounded by the two of the four known value lattice points but is not at a vertex of the tetrahedron, the two of the four known lattice points being A and B with values a and b, the value p returned is given by:

p=((Distance(AP).b)+(Distance(BP).a))/Distance(AB).

9. (*Currently amended*) The method as claimed in claim **1**, wherein the known value lattice points form a sparse lattice with known value lattice points separated from each other by an integer multiple of the distance between adjacent lattice points.

10. (*Currently amended*) The method as claimed in claim **9**, wherein said integer multiple is an integer power of **[[**two**]]** 2.

11. (*Previously presented*) The method as claimed in claim **10**, wherein the integer is 4 and all given lattice points coincide with a value lattice point or lie between two adjacent value lattice points or lie within a triangle described by three adjacent value lattice points.

12. (*Previously presented*) The method as claimed in claim **10**, wherein the integer is 8 or more and all given lattice points coincide with a value lattice point or lie between two adjacent value lattice points or lie within a triangle described by three adjacent value lattice points or lie within or lie within a tetrahedron of four adjacent value lattice points.

13. (*Currently amended*) The method as claimed in claim **12**, ~~where~~ wherein the integer is 8.

14. (*Previously presented*) The method as claimed in claim **2**, wherein the step of establishing a value comprises determining a set of four known value lattice points which form a tetrahedron touching or enclosing the given lattice point, and providing the weighted average from the positions of four known value lattice points, the known values of one or more of the four known value lattice points, and the position of the given lattice point.

15. (*Previously presented*) The method as claimed in claim **14,** wherein the step of providing the weighted average comprises using the positions as inputs to a jump table.

16. (*Currently amended*) A method of mapping values in a first color space to values in a second color space, comprising:

establishing **[[**the**]]** a value in the second color space;

establishing an n-dimensional lattice having a plurality of lattice points, the ~~function~~ second color space having values at the lattice points, wherein n is a positive integer greater than or equal to **[[**two**]]** 2;

recording values for a subset of the lattice points, the lattice points of the subset being known value lattice points; and

establishing a value for a given lattice point by determining the values of only m of (n+1) known value lattice points defining an ~~n-simples~~ n-simplex touching or enclosing the given lattice point, wherein m is a positive integer equal to the number of n-simplexes of non-zero volume whose vertices consist of the given lattice point and n of the (n+1) known value lattice points, and by returning a weighted average of said m of the known value lattice points.

17. (*Currently amended*) A computer programmed to determine a value for a function, by:

establishing an n-dimensional lattice having a plurality of lattice points, the function having values at the lattice points, **[[**and**]]** where n is greater than or equal to **[[**two**]]** 2;

recording values for a subset of the lattice points, the lattice points of the subset being known value lattice points; and

establishing a value for a given lattice point by determining the values of only m of (n+1) known value lattice points defining an n-simplex touching or enclosing the given lattice point, wherein m is a positive integer equal to the number of n-simplexes of non-zero volume whose vertices consist of the given lattice point and n of the (n+1) known value lattice points, and by returning a weighted average of said m of the known value lattice points.

18. (*Previously presented*) The computer as claimed in claim **17,** wherein n=3 and the n-simplex comprises a tetrahedron.

19. (*Currently amended*) The computer as claimed in claim **18,** wherein the computer is programmed such that:

a weighted average of all four known value lattice point values is used if the given lattice point is enclosed by the tetrahedron but is not touched by a face of the tetrahedron,

a weighted average of three of the four known value lattice point values is used if the given lattice point is on a face of the

tetrahedron bounded by the three of the four known value lattice points but is not touched by an edge of the tetrahedron,

a weighted average of two of the four known value lattice point values is used if the given lattice point is on an edge of the tetrahedron bounded by the two of the four known value lattice points but is not at a vertex of the tetrahedron, and ~~wherein~~

a value of one of the known value lattice points is used if the given lattice point is also the known value lattice point.

20. (*Currently amended*) A program storage medium readable by a computer, tangibly embodying a program of instructions executable by the computer to perform method steps for determining a value for a function, said method steps comprising:

establishing an n-dimensional lattice having a plurality of lattice points, the function having values at the lattice points, ~~and where~~ wherein n is greater than or equal to **[[**two**]]** 2;

recording values for a subset of the lattice points, the lattice points of the subset being known value lattice points; and

establishing a value for a given lattice point by determining the values of only m of (n+1) known value lattice points defining an n-simplex touching or enclosing the given lattice point, wherein m is a positive integer equal to the number of n-simplexes of non-zero volume whose vertices consist of the given lattice point and n of the (n+ 1)

known value lattice points, and by returning a weighted average of said m of the known value lattice points.

21. (*Previously presented*) The program storage medium as claimed in claim **20**, wherein n=3 and the n-simplex comprises a tetrahedron.

22. (*Currently amended*) The program storage medium as claimed in claim **21**, wherein in the step of establishing a value**[[,]]**<u>:</u>

a weighted average of all four known value lattice point values is used if the given lattice point is enclosed by the tetrahedron but is not touched by a face of the tetrahedron,

a weighted average of three of the four known value lattice point values is used if the given lattice point is on a face of the tetrahedron bounded by the three of the four known value lattice points but is not touched by an edge of the tetrahedron,

a weighted average of two of the four known value lattice point values is used if the given lattice point is on an edge of the tetrahedron bounded by the two of the four known value lattice points but is not at a vertex of the tetrahedron, and ~~wherein~~

a value of one of the known value lattice points is used if the given lattice point is also the known value lattice point.

23. (*Cancelled*)

24. (*New*) The method as claimed in claim **1**, wherein the (n+1) known lattice points enclose the given lattice point such that the given lattice point is not coincident with any of the known value lattice points, the known value lattice points do not define an n-simplex touching the given lattice point, and the given lattice point is not on a diagonal between any known value lattice points of the n-simplex.

25. (*New*) The computer as claimed in claim **17**, wherein the (n+1) known lattice points enclose the given lattice point such that the given lattice point is not coincident with any of the known value lattice points, the known value lattice points do not define an n-simplex touching the given lattice point, and the given lattice point is not on a diagonal between any known value lattice points of the n-simplex.

26. (*New*) The medium as claimed in claim **20**, wherein the (n+1) known lattice points enclose the given lattice point such that the given lattice point is not coincident with any of the known value lattice points, the known value lattice points do not define an n-simplex touching the given lattice point, and the given lattice point is not on a diagonal between any known value lattice points of the n-simplex.

27. (*New*) The method as claimed in claim **15**, further including:

listing interpolation equations that are applicable for a particular case associated with the n-dimensional lattice;

responding to digital values representing the m known value lattice points such that high order bits of the digital values provide a table offset that determines which coarse lattice is to be used and low order bits of the digital values choose which case to execute, wherein each case corresponds to an intermediate point between the m known value lattice points; and

causing each chosen case to access the values for the specific coarse lattice points required by the interpolation equations for that case.